Anatomy of a microcomputer
Commodore PET

User port

Pin Identification Character	Signal Label	Signal Description
1	Ground	Digital Ground
2	T.V. Video	Video output used for external display, used in diagnostic routine for verifying the video circuit to the display board.
3	IEEE-SRQ	Direct connection to te SRQ signal on the IEEE-488 port. It is used in verifying operation of the SRQ in the diagnostic routine.
4	IEEE-EOI	Direct connection to the EOI signal on the IEEE-488 port. It is used in verifiyg operation of the EOI in the diagnositc routine.
5	Diagnostic Sense	When this pin ie held low during power up the PET software jumps to the diagnostic routine, rather than the BASIC routine.
6	Tape #1 READ	Used with the diagnostic routine to verify cassette tape #1 read function.
7	Tape #2 READ	Used with the diagnostic routine to verify cassette tape #2 read function.
8	Tape Write	Used with the diagnostic routine to verify operation of the WRITE function of both cassette ports.
9	TV Vertical	TV vertical sync signal verified in diagnostic. May be used for external TV display.
10	TV Horizontal	TV horizontal signal verified in diagnostic may be used for TV display.
11, 12	GND	Digital ground
A	GND	Digital ground
B	CA1	Standard edge sensitive input of 6522VIA.
C	PA0	
D	PA1	
E	PA2	Input/output lines to peripherals and can be programmed independently of each other for input or output
F	PA3	
H	PA4	
J	PA5	
K	PA6	
L	PA7	
M	CB2	Special I/O pin of VIA.
N	GND	Digital ground.

Arithmetic functions

FUNCTION	EXAMPLE	PURPOSE
ABS	10 C = ABS(A)	Returns magnitude of argument without regard to sign.
ATN	10 C = ATN(A)	Returns arctangent of argument, C will be expressed in radians.
COS	10 C = COS(A)	Returns cosign of argument. A must be expressed in radians.
DEF FN	10 DEF FNA(B) = C*D	Allows user to define a function. Function label A must be a single letter; argument B is a dummy.
EXP	10 C = EXP(A)	Returns constant 'e' raised to power of the argument. In this example, e^A
INT	10 C = INT(A)	Returns largest integer less than or equal to argument.
LOG	10 C = LOG(A)	Returns natural logarithm of argument. Argument must be greater than or equal to zero.
RND	10 C = RND(A)	Generates a random number between zero and one. If A is less than O, the same negative random number is produced in each call to RND. If A = O, the same positive sequence of random numbers is generated each time RND is called. If A is greater than O, a new sequence is produced for each call to RND.
SGN	10 C = SGN(A)	Returns −1 if argument is negative, returns 0 if argument is zero, and returns +1 if argument is positive.
SIN	10 C = SIN(A)	Returns sine of argument. A must be expressed in radians.
SQR	10 C = SQR(A)	Returns square root of argument.
TAN	10 C = TAN(A)	Returns tangent of argument. A must be expressed in radians.

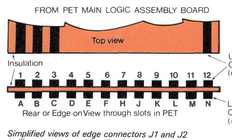
Arithmetic operators

SYMBOL	EXAMPLE	PURPOSE
=	10 A = B 20 LET A = B	Assigns a value to a variable Let is optional
↑	20 PRINT A ↑ 2	Exponentiation; in example, A^2.
/	35 C = A/8	Division
*	40 C = A*8	Multiplication
+	50 C = A + 8	Addition
−	60 C = A − 8	Subtraction
=	10 IF A = B THEN PRINT C	A 'equals' B
<>	10 IF A<>B THEN C = 4	A 'does not equal' B
<	10 IF A<B THEN C$ = "X"	A 'is less than' B
>	10 IF A>B THEN C$ = DX	A 'is greater than' B
<=	10 IF A<=B THEN C = 20	A 'is less than or equal to' B
>=	10 IF A>=B THEN C = D − 1	A 'is greater than or equal to' B
AND	10 IF A<D AND B then C=40	A and B must BOTH be true for statement 10 to be true.
OR	20 IF A OR B the C = 90	A must be true or B must be true for statement 20 to be true.
NOT	30 IF NOT A THEN PRINT C	Expression is true if A is false.

IEEE connector

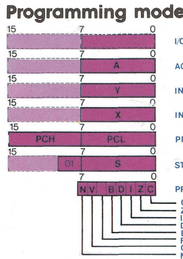
PET Pin Characters	Standard IEEE Connector pin numbers	IEEE Signal Mnemonic	Signal Definition/Label
UPPER PINS			
1	1	D101	Data input/output line #1
2	2	D102	Data input/output line #2
3	3	D103	Data input/output line #3
4	4	D104	Data input/output line #4
5	5	EOI	End or identity
6	6	DAV	Data valid
7	7	NRFD	Not ready for data
8	8	NDAC	Data not accepted
9	9	IFC	Interface clear
10	10	SRQ	Service request
11	11	ATN	Attention
12	12	GND	Chassic ground and IEEE cable shield drain wire
LOWER PINS			
A	13	D105	Data input/output line #5
B	14	D106	Data input/output line #6
C	15	D107	Data input/output line #7
D	16	D108	Data input/output line #8
E	17	REN	Remote enable
F	18	GND	DAV ground
LOWER PINS			
H	19	GND	NRFD ground
J	20	GND	NDAC ground
K	21	GND	IFC ground
L	22	GND	SRQ ground
M	23	GND	ATN ground
N	24	GND	Data ground (D101-8)

Pet interfaces and lines


FROM PET MAIN LOGIC ASSEMBLY BOARD
Top view
Insulation
1 2 3 4 5 6 7 8 9 10 11 12
A B C D E F H J K L M N
Upper Contact (or pin)
Lower Contact (or pin)
Rear or Edge on View through slots in PET

Simplified views of edge connectors J1 and J2 to illustrate contact identification convention

Programming model MCS650X



Solid line indicates currently available features
Dashed line indicates forthcoming members of family

KEYBOARD

SCREEN (40 CHAR x 25)

POWER TRANSFORMER

IEEE-488 PORT

USER PORT

C2N PORT

6502 MICROPROCESSOR CHIP

6522 VERSATILE INTERFACE ADAPTOR (VIA)

6520 PERIPHERAL INTERFACE ADAPTOR (PIA)

ROMS (READ ONLY MEMORY)

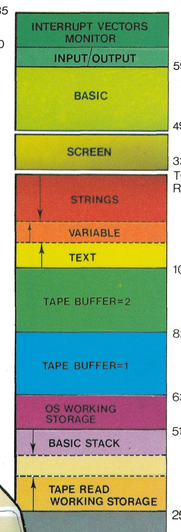
MEMORY EXPANSION PORT

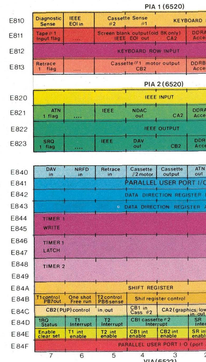
CHARACTER GENERATOR ROM

VIDEO CONNECTOR

RAMS (RANDOM ACCESS MEMORY)

Memory map





MCS6501-MCS6505 Microprocessor instruction set

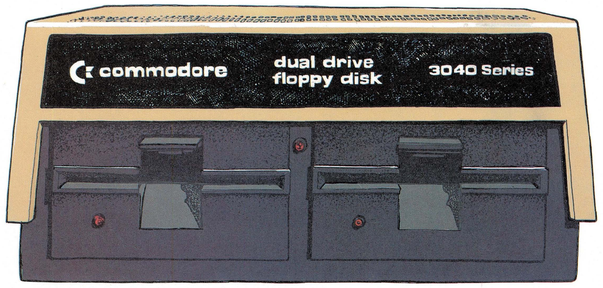
ADC	Add Memory to Accumulator with Carry	JSR	Jump to New Location Saving Return Address
AND	"AND" Memory with Accumulator		
ASL	Shift Left One Bit (Memory or Accumulator)	LDA	Load Accumulator with Memory
		LDX	Load Index X with Memory
BCC	Branch on Carry Clear	LDY	Load Index Y with Memory
BCS	Branch on Carry Set	LSR	Shift Right One Bit (Memory or Accumulator)
BEQ	Branch on Result Zero		
BIT	Test Bits in Memory with Accumulator	NOP	No Operation
BMI	Branch on Result Minus		
BNE	Branch on Result not Zero	ORA	"OR" Memory with Accumulator
BPL	Branch on Result Plus		
BRK	Force Break	PHA	Push Accumulator on Stack
BVC	Branch of Overlow Clear	PHP	Push Processor Status on Stack
BVS	Branch on Overflow Set	PLA	Pull Accumulator from Stack
		PLP	Pull Processor Status from Stack
CLC	Clear Carry Flag		
CLD	Clear Decimal Mode	ROL	Rotate One Bit Left (Memory or Accumulator)
CLI	Clear Interrupt Disable Bit	ROR	Rotate One Bit Right (Memory or Accumulator)
CLV	Clear Overflow Flag	RTI	Return from Interrupt
CMP	Compare Memory and Accumulator	RTS	Return from Subroutine
CPX	Compare Memory and Index X		
CPY	Compare Memory and Index Y	SBC	Subtract Memory from Accumulator with Borrow
		SEC	Set Carry Flag
DEC	Decrement Memory by One	SED	Set Decimal Mode
DEX	Decrement Index X by One	SEI	Set Interrupt Disable Status
DEY	Decrement Index Y by One	STA	Store Accumulator in Memory
		STX	Store Index X in Memory
EOR	"Exclusive-Or" Memory with Accumulator	STY	Store Index Y in Memory
		TAX	Transfer Accumulator to Index X
INC	Increment Memory by One	TAY	Transfer Accumulator to Index Y
INX	Increment Index X by One	TSX	Transfer Stack Pointer to Index X
INY	Increment Index Y by One	TXA	Transfer Index X to Accumulator
		TXS	Transfer Index X to Stack Pointer
JMP	Jump to New Location	TYA	Transfer Index Y to Accumulator

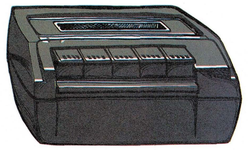
Basic commands and statements

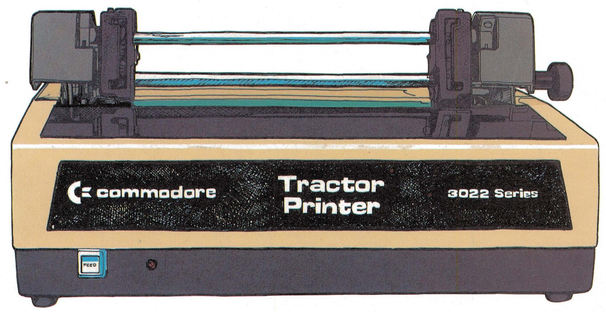
COMMAND/ STATEMENT	EXAMPLE	PURPOSE
CLR	CLR	Sets variables to zero or null
CMD	CMD D	Keep IEEE device D open to monitor bus
CONT	CONT	Continue program execution after a STOP command. No program changes allowed
GOTO	GOTO L	Continue program execution at line L after a STOP command. Program changes are permitted.
FRE	PRINT FRE (0)	Returns number of bytes of available memory
END	END	Ends program
LET	LET A = 2	Assign a value to a variable
LIST	LIST LIST −L LIST L-M LIST L-	Lists current program List current program through line L Lists lines L through M of current program Lists current program from line L to end.
LOAD	LOAD	Loads next encountered program from built-in tape unit
	LOAD "NAME" LOAD "NAME",D	Loads program NAME from built-in tape unit Loads program NAME from device D.
NEW	NEW	Deletes current program from memory, sets variables to zero
PEEK	PEEK(A)	Returns byte value from address A.
POKE	POKE A,B	Loads byte B into address A
PRINT	PRINT A PRINT A$ PRINT # L,A PRINT # L,A$	Prints value of A on display screen Prints specified string on screen Prints value of A on logical file L Prints specified string on logical file L
RUN	RUN	Begins execution of program at lowest line number
SAVE	RUN L SAVE SAVE "NAME" SAVE "NAME",D SAVE "NAME",D,C	Begins execution of program at line L Saves current program on built-in tape unit Saves current program NAME on built-in tape unit Saves current program NAME on device D Saves program NAME on device D. C specifies EOF or EOT
STOP	STOP	Stops program execution
SYS	SYS (X)	Complete control of PET is transferred to a subsystem at decimal address contained in the argument.
TI$	TI$ = "HHMMSS"	Set PET's internal clock to real time
TI	PRINT TI	Displays number of 'jiffies' since PET was powered up or clock was zeroed (A jiffy = 1/60 of a second)
USR	USR(X)	Transfers program control to a program whose address is at locations 1 and 2. X is a parameter passed to and from the machine language monitor.
WAIT	WAIT A,B,C	Stops execution of BASIC until contents of location A, ANDed with B and exclusive ORed with C, is not equal to zero. C is optional and defaults to zero.
CLOSE	10 CLOSE L	Closes logical file L
DATA	10 DATA 1,2,3,4 20 DATA TOM,SUE 30 DATA "TOM DOE"	Specifies data to be read from left to right Alphabetics do not need to be enclosed in quotes If strings contain spaces, commas, colons, or graphic characters, the string must be enclosed in quotes
DIM	10 DIM A(n) 20 DIM A(n,m,o,p) 30 DIM A(n),B(m) 40 DIM A(N) 50 DIM A$(n)	Spedifies maximum number of elements in an array or matrix Specifies maximum number of dimensions in an array Number of arrays limited by memory May be dimensioned dynamically Strings may be dimensioned
END	999 END	Terminates program execution
GET	10 GET C 20 GET C$ 30 GET #L,C 40 GET #L,C$	Accepts single numeric character from keyboard Accepts single string character from keyboard Accepts single character from specified logical file Accepts specified single string character from logical file
INPUT	10 INPUT A 20 INPUT A$ 30 INPUT A,A$,B,B$ 40 INPUT #L,A 50 INPUT #L,A$ 60 INPUT #L,A,A$,B,B$	Accepts value of A from keyboard Accepts value of string variable A from keyboard. The string does not have to be enclosed in quotes Accepts specified string values from keyboard Accepts value of A from logical file L Accepts specified string from logical file L Accepts specified values and strings from logical file L. Strings do not have to be enclosed in quotes.
LOAD	10 LOAD 20 LOAD "NAME"	Loads next encountered program or file, on built-in tape unit, into PET's memory. Loads program or file NAME into memory from built-in tape unit.
	20 LOAD "NAME",D	Loads specified NAME from device D.
OPEN	10 OPEN L	Opens logical file L for read only from tape unit.
	20 OPEN L,D 30 OPEN L,D,C	Opens logical file L for read only from device D Opens logical file L for command C from device D.
	40 OPEN L,D,C, "NAME"	Opens logical file L on device D. If device D accepts formatted files, file NAME is positioned for command
POS	10 PRINT POS(0)	Prints next available print position (position of cursor on screen)
PRINT	10 PRINT A 20 PRINT A$ 30 PRINT A,A$	Prints value of A on display screen. Prints specified string on screen. Prints specified values of strings on screen, beginning in next available print position (pre-TABbed positions are in columns 10,20,30,40 etc).
	40 PRINT A;A$	Prints on specified vlues and strings on screen separated by 3 spaces if numeric, concatenated of string.
	50 PRINT #L,A 60 PRINT #L,A$	Prints specified value to logical file L. Prints specified string to logical file L.
READ	10 READ A 20 READ A$ 30 READ A,A$,B,B$	Obtains value of A from DATA statement. Obtains string value of A from DATA statement. Obtains specified values for strings and numeric variables from DATA statements.
REM	10 REM **COMMENT**	Inserts non-executable comments in a program for documentation purposes.
RESTORE	10 RESTORE	Permits re-reading of DATA statements without re-running program.
TAB	10 PRINT TAB(N);A 20 PRINT TAB(N);A$	Prints value of A in character position N + 1 on screen. Prints string beginning in character position N + 1 on screen.
VERIFY	10 VERIFY	Verifies most recent program saved on cassette by reading it and comparing it with program still in PET's memory.
	20 VERIFY "NAME"	Verifies specified program NAME saved on built-in cassette by reading it and comparing it with program still in PET's memory.
	30 VERIFY "NAME",D	Verifies specified program NAME saved on device D by reading it and comparing it with program still in PET's memory.
SPC	10 PRINT (SPC)N	Prints N spaces or blanks.
FOR . . NEXT	10 FOR A = 1 TO 20 . . 90 NEXT A	Loop control. Performs all functions between FOR and NEXT as many times as specified by index. In this example, the index variable is A.
STEP	10 FOR A = 1 TO 20 STEP 2 . . 90 NEXT A	Step specifies size of increment to be added to index to increase or decrease its value towards the desired number of iterations.
IF . . THEN	10 IF A = 10 THEN PRINT A	If condition is 'TRUE', instruction following 'THEN' (in this example, 'PRINT A') would be executed. Otherwise, the next statement is sequence is executed.
IF . . GOTO	10 IF A = 1 GOTO L	If condition is true, control is transferred to specified line. Otherwise, the next statement, following the IF . . GOTO, is executed.
GOTO	10 GOTO L	Transfers control (jumps) to specified line, skipping over intervening lines.
GOSUB	10 GOSUB L	Begins execution of a subroutine which begins on a specified line.
ON . . GOTO	10 ON A GOTO L,M,N	Transfers control to specified line (in this example L, M or N. depending on value of index A.
ON . . GOSUB	10 ON A GOSUB L,M,N	Begins execution of subroutine which begins on line L,M or N, depending on the value of index A.
RETURN	9990 RETURN	Subroutine exit; transfers control to the statement following most recent GOSUB directing transfer to the subroutine.
ASC	10 A = ASC("XYZ")	Returns integer value corresponding to ASCII code of first character in string.
CHR$	10 A$ = CHR$(N)	Returns character corresponding to ASCII code number.
LEFT$ LEN MID$	10 ?LEFT$(X$,A) 10 ?LENX$) 10 ?MID$(X$,A,B)	Returns leftmost A characters from string. Returns length of string. Returns B characters from string, starting with the Ath character.
RIGHT$ STR$ VAL	10 ?RIGHT$(X$,A) 10 A$ = STR$(A) 10 A = VAL(A$) 20 A = VAL("A")	Returns rightmost A characters from string Returns string representation of number. Returns numeric representation of string. If string not numeric, returns "0".

Monitor commands

M	Display memory
R	Display register
G	Begin execution
X	Exit to BASIC
L	Load
S	Save







specifications

Dual drive floppy disk unit 3040

Microcomputer system devices
Controller
6504 Microprocessor
6530 I/O RAM, and 1K ROM software.
6316 2K ROM for encoding and decoding disk data.
6522 I/O and interval timers
File interface.
6502 microprocessor.
(2) 6532 RAM, I/O, interval timers
(2) 6332 4K ROM each (total 8K disk operating system). Shared RAM.

Disk Drives: (2) Shugart Associates SA390 drives standard minifloppy (5¼" disk).
Activity LEDs light when a file is open on that drive.
Error LED indicates command or data error.

Packaging
18 gauge all steel cabinet.

Dimensions: Width: 15"; depth: 14.35"; height: 6.5".
Cover hinges from base for servicing.
Weight: 28lbs.

Diskette Organisation
Formatting is by the drive itself — any mini-floppy diskette may be used.
35 concentric tracks.
Constant density recording on each track varying number of sectors per track: — Innermost: 17; outermost: 21.
176640 bytes on a single side.
Track 18 used for directory.
171520 bytes for user storage.
Soft-sectoring.

Data Interface
IEEE-488.
Standard 24-pin stacking connector. Device 8-15 by jumper option
Full listener-talker recognises secondary addressing.

Floppy Disk Commands (summary)
LOAD program
SAVE program
VERIFY program
OPEN datafile (up to 5)
PRINT # to data file
CLOSE data file
VALIDATE disk
DUPLICATE disk
COPY program or data file (allows data files to be concatenated
RENAME disk
SCRATCH unwanted file
examine ERROR CHANNEL

Advanced Disk Commands (summary)
BLOCK-READ
BLOCK-WRITE
BLOCK-EXECUTE
BLOCK-POINTER
BLOCK-ALLOCATE
BLOCK-FREE
memory write
memory read
memory execute
USER

Pet computer

Dimensions: 16½" wide by 18½" deep. 14" overall height.
Weight: 44lbs.

MEMORY
Random Access Memory 8K, 16K and 32K as per model number. Expandable to 32K bytes
Read Only Memory (operating system resident in the computer): 13K bytes
8K BASIC interpreter
5K Operating system
1K Machine code monitor
Video Display Unit: 9" enclosed black and green, high resolution CRT.
1000 character display, arranged 40 columns by 25 lines.
8 x 8 dot matrix for characters and continuous graphics.
Automatic scrolling from bottom of screen.
Winking cursor with full motion control.
Reverse field on all characters (white on black or black on white)
64 standard ASCII characters; 64 graphic characters. Lower case letters are available by use of a POKE command.

KEYBOARD
73 keys. All 64 ASCII characters available without shift. Calculator style numeric keypad. All 64 graphic and reverse field characters accessible from keyboard (with shift).
Screen Control: Home and clear
Editing: Character insertion and deletion.

OPERATING SYSTEM
Powerful and 'user friendly' screen editor. Machine language accessibility.
File management in operating system. Cursor control, reverse field and graphics under simple BASIC control. Real time clock can be accessed from BASIC.
File management in operating system can be handled by BASIC or from machine code.

INPUT/OUTPUT
All other I/O supported through IEEE-488 instrument interface which allows for multiple intelligent peripherals. All I/O automatically managed by operating system software. Single character I/O with GET command. Easy screen line-edit capability allows for BASIC expansion with up to 12 intelligent peripherals. Direct sensing of key press available.

BASIC INTERPRETER
Expanded 8K BASIC:
Upward expansion from current popular BASIC language.
Strings, integers and multiple dimension arrays.
10 significant digits; floating point numbers.
Memory access through PEEK and POKE commands.
Machine code subroutines can be called up from BASIC with SYS and USR.
Pseudo random number generator

Magnetic cassette unit C2N

Specifications:
Fast Commodore designed redundant-recording scheme, assuring reliable program and data storage. Cassette drive designed by Commodore for high reliability of recording and data retention.
High noise immunity, error detection and correction.
Named tape files.
Verify function checks program accuracy.

TAPE COMMANDS (Summary)
LOAD program
SAVE program
VERIFY program
OPEN data file
PRINT # to data file
CLOSE data file

Tractor feed printer 3022

Microcomputer system devices
6504 Microprocessor
6532 I/O RAM, inverval timer (2)
6332 4K x 8 ROM

Printer Mechanism: Tractor feed
Epson DH-70 print head.
Dot matrix — 7 x 6 — 80 columns per line.
Impact print — original plus 3 copies.
Print rate is 75 LPM (93 CPS).
Programmable line spacing
Forms: 8.5 + plus .5+ x 2 (sprocket margins).
Pin to pin distance:
.5+ Longitudinally
9.0+ Laterally
5/32+ Diameter

Packaging:
18 gauge all steel cabinet. Dimensions: width: 17", depth: 18", height: 6½".
Forms enter from rear or bottom of cabinet. Weight: 18lbs.

Data Interface:
IEEE-488
Standard 24-pin stacking connector. Device 4-11 by jumper option. Listener only. Recognises secondary addressing.

Character Set:
Upper Case ASCII
Lower Case ASCII
PET Graphics

Control Characters:
- Enhance printing (doubles size)
- Enable automatic line count and paging
- page eject
- Print reverse field
- Overprint a line
- Switch to lower case character set
- Print programmable character
IEEE Secondary Address Commands
- Print data exactly as received
- Accept characters as a format
- Edit data to format
- Alter number of lines per page
- Enable diagnostic messages to print
- Accept data for programmable character
- Alter paper advance per line feed

Data Formatting Capability
- Field width and decimal position specified
- Leading or trailing sign
- Fixed or floating dollar sign
- Forced leading zeros
- Literal characters always printed
- Alpha fields left justified

Diagnostic Messages
- Can be turned on when desired
- Print on paper
- Describe problems with format and data.

Headquarters: Commodore Electronics Limited, Saasoon House, Shirley and Victoria, P.O. Box N-10256, Nassau, Bahamas.
USA: Commodore Business Machines Inc., 3330 Scott Boulevard, Santa Clara, California 95050.
Canada: Commodore Business Machines Limited, 3370 Pharmacy Avenue, Agincourt, Ontario M1W 2K4, Canada.

European Office: Commodore Electronics Limited, Dufourstrasse 9, Postfach 4010 Basel, Switzerland.
United Kingdom: Commodore Systems Division (UK) Limited, 818 Leigh Road Trading Estate, Slough, Berkshire.
West Germany: Commodore Buercmachinen GmbH, Dornhofstrasse 38, 6078 Neu-Isenburg.

Switzerland: Commodore A.G. Schweiz, Dufourstrasse 9, 4010 Basel.
Japan: Commodore Japan Limited, Taisei-Denshi Building, 8-14 Ikue, 1-Chome, Asahi-ku, Osaka 535.
Australia: Commodore, P.O. Box 336, 3 Campell Street, Artarmon, New South Wales 2064.